

性能測定条件・結果として 明示すべき項目について

クーガー株式会社
リードプロジェクトマネジャー 清水啓太

ETHTerakoyaでは今年6月以降、7回の有識者ワークショップを通じてブロックチェーンの性能評価指標についてディスカッションを行ってきた

スケジュール


■過去のディスカッション内容は以下のとおり

#	日程	内容
1	2021/6/3	・ ワーキンググループの目的・進め方・成果物について
2	2021/6/21	・ ブロックチェーンの性能評価にあたり重要な指標について
3	2021/7/7	・ 性能評価指標の深掘り
4	2021/8/5	・ テスト条件および実施方法について
5	2021/8/25	・ テスト実施・結果共有と検知した課題の共有1
6	2021/9/16	・ テスト実施・結果共有と検知した課題の共有1
7	2021/10/1	・ テスト実施・結果共有と検知した課題の共有2
8	2021/10/29	・ 公開ワークショップ

参加企業

■ブロックチェーン開発に実績を持つ企業、また業務利用を検討・開始している企業でワーキンググループを形成し、ディスカッションを進めた

HITACHI
Inspire the Next

 日立ソリューションズ

NTT DATA

 **NTTテクノクロス**

 **accenture**

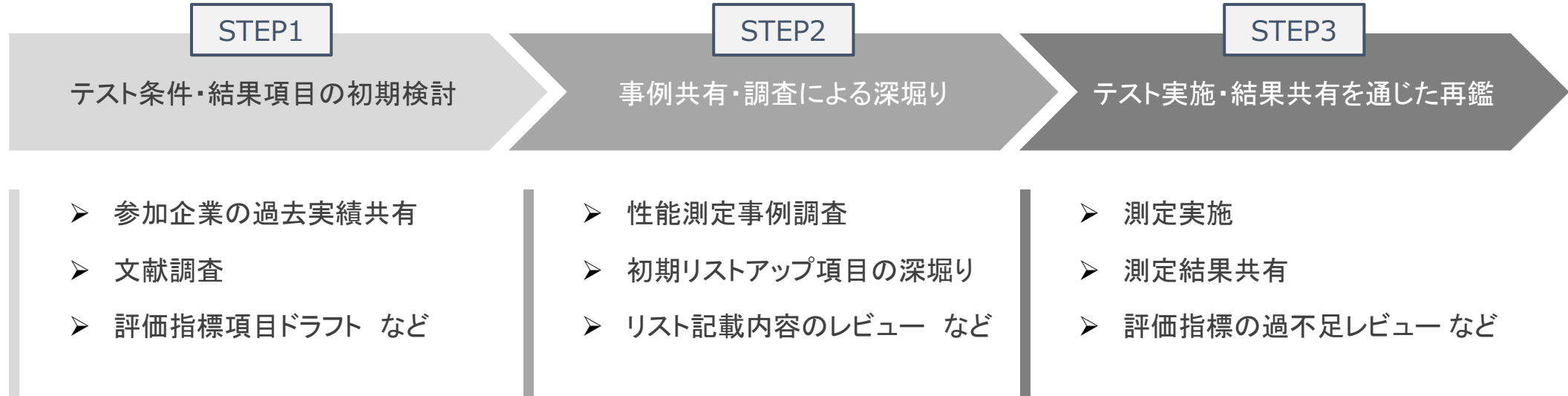
野村アセットマネジメント

 **SingulaNet**

COMPS

性能検証の際に注視すべきテスト条件・結果項目については、初期的にリストアップした項目に対し、参加企業それぞれの過去事例・調査や検証をもとに肉付けを行うことで最終化を進めた

性能評価指標の精査プロセス



テスト条件・結果項目ごとに3つの観点に分類し、下位項目にブレイクダウンする方法をとった結果項目に対するハードウェア情報は存在しないため、残る5分類について詳述する

テスト条件・結果項目の細分化

		観点		
		チェーンに関する情報	ハードウェア情報	システム負荷条件
入出力	テスト条件	● 対象	● 対象	● 対象
	テスト結果	● 対象	— 非対象	● 対象

テスト条件としてチェーンの情報は必須である。クライアント名のほか、合意形成アルゴリズムの別や、対象とするメソッド、ノード数まで条件とすることで公平なチェーンの比較が可能となる

テスト条件項目1 - チェーンに関する情報

種別	項目	定義	例
チェーンに関する情報	Blockchain Client Name	ブロックチェーンのクライアント名	Ethereum / Geth, Hyperledger Fabricなど
	Consensus Algorithm	ブロックチェーンにおけるネットワーク全体の合意を行うための方式	Proof of Work (Ethash), Proof of Authority (Clique) など
	Transaction Method	ブロックチェーンのデータがある値から別の値に変更する状態遷移法。性能測定の対象となる処理の内容	性能評価に使用したスマートコントラクトのコード など
	Network Size(Node数)	合意形成に参加しているバリデータノードの数	ノード台数

ハードウェア性能はチェーンのパフォーマンスにも影響する要素であるため、チェーンの動作環境、および負荷生成サーバの環境についても情報を提示することが望ましい

テスト条件項目2 - ハードウェア情報

種別	環境	項目	定義	例
ハードウェア 情報	クラウドの 場合	使用したクラウドサービス	クラウドサービス名	AWS, GCP, Azureなど
		インスタンスタイプ	クラウドサービスのインスタンス名	2.xlarge, t3.mediumなど
		RAM容量	クラウドサービスのインスタンスのRAM容量	16GB
	実マシンの 場合	CPU種別	マシンのCPU種別	Intel Core i9 3.5GHz など
		GPU種別 ※ PoW に GPU を 使う場合	マシンのGPU種別	RTX 3090など
		RAM 容量	マシンのRAM容量	16GB

ツールを用いてチェーンにかける負荷についても、テスト結果に影響を及ぼすため、同様に条件を記載する必要がある

テスト条件項目3 - システム負荷条件

種別	項目	定義	例
システム 負荷条件	負荷生成クライアント数	システム負荷をかけたクライアントの数	200
	負荷総リクエスト数	クライアントから送信した総リクエスト数	73,555 tps
	負荷継続時間	クライアントからかけた負荷の継続時間	60秒
	負荷ツール単体での 限界リクエスト数	負荷ツール単体でかけられるリクエスト数の最大値	13,555 tps

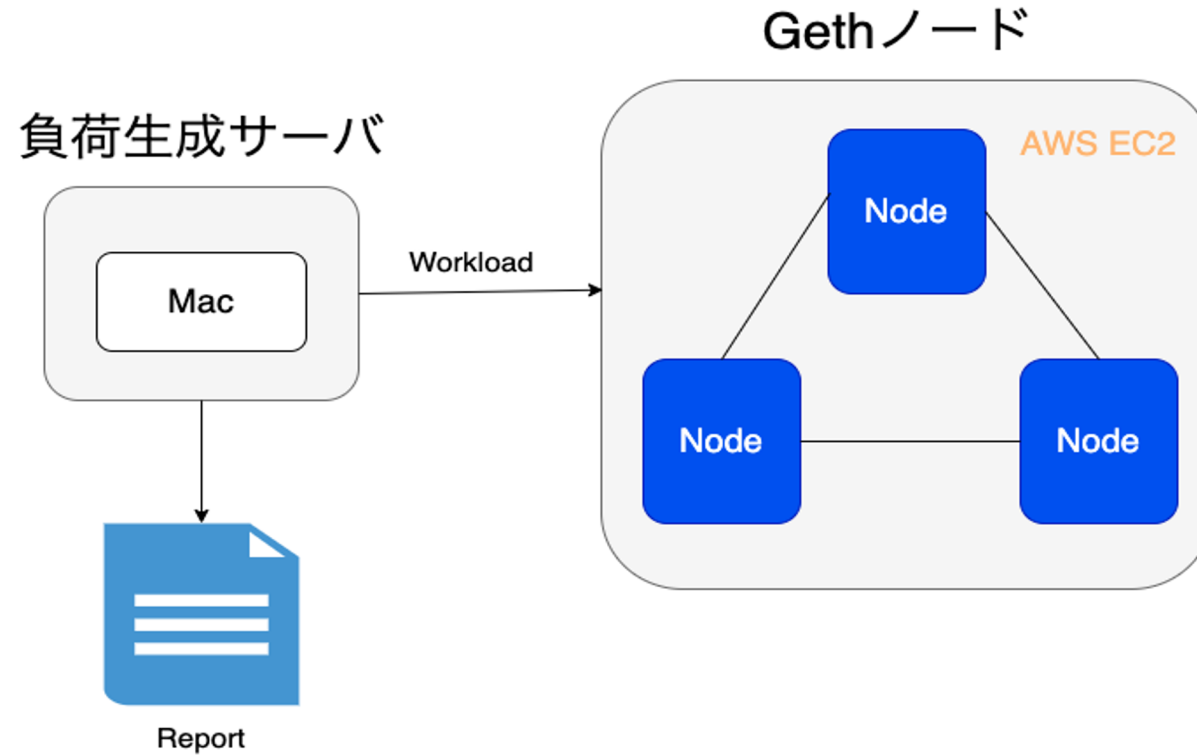
テスト結果としては、処理に対するレイテンシ・スループット、およびCPUやディスクへの負荷を見ることでチェーンの性能を見ることができると考えられる

テスト結果項目

種別	項目	定義	値(例)
ブロックチェーンに関する情報	Read Latency	読み取り要求を送信し、その応答を受信するまでの合計時間	0.18 秒
	Read Throughput	1秒あたりの読み取り数	813.1
	Transaction Latency	ネットワーク全体がトランザクションを検証するのにかかる時間	11.18 秒
	Transaction Throughput	定義された期間にブロックチェーンによって有効なトランザクションがコミットされる割合	27.4
システム負荷情報	CPU負荷	CPUに対する負荷	Max: 56.7% Avg: 21.83%など
	ディスク容量負荷	ディスク容量に対する負荷	Min:0.80 KB/s Max:146.80 KB/s Avg:49.47 KB/s

実際に弊社でも各テスト条件・結果項目の明示を前提として、ノード数3台のGethネットワークを構築し性能検証を行った

システム構成概要図



ワーキンググループを通じて策定した条件・結果項目に相当する情報は以下のとおり

テスト実施条件 - チェーンに関する情報

- 今回クライアントはGethを使用
- メソッドはtransactionの測定のためにTransferを、readの測定のためにbalanceOfを対象とした

項目	定義	値
Blockchain Client Name	ブロックチェーンのクライアント名	Geth
Consensus Algorithm	ブロックチェーンにおけるネットワーク全体の合意を行うための方式。	Proof of Authority (Clique)
Transaction Method	ブロックチェーンのデータある値から別の値に変更する状態遷移法。性能測定の対象となる処理の内容。	Transfer balanceOf
Network Size (Node数)	合意形成に参加しているバリデータノードの数。	3台

テスト実施条件 - ハードウェア情報

- 負荷生成はMacPCを利用、チェーンのノードはAWSを用いてネットワークを構成した

項目	定義	値
負荷生成サーバ: 実マシン		
CPU 種別	マシンのCPU種別	Apple M1
GPU種別 ※ PoW に GPUを 使う 場合	マシンのGPU種別	Apple独自設計 8コア
RAM容量	マシンのRAM容量	16 GB
ブロックチェーンの各ノード: クラウド		
使用したクラウドサービス	クラウドサービス名	AWS
インスタンスタイプ	クラウドサービスのインスタンス名	t2.xlarge
RAM容量	クラウドサービスのインスタンスのRAM容量	16 GB

負荷の総リクエスト数は1000トランザクションとし、検証を実施 個別条件として目標TPSは50TPS、block生成時間は10秒、成功とみなす承認数は2ブロックとした

テスト実施条件 - システム負荷条件

■総リクエストは1,000に設定

項目	定義	値
負荷生成クライアント数	システム負荷をかけたクライアントの数	1
負荷総リクエスト数	クライアントから送信した総リクエスト数	1,000トランザクション数
負荷継続時間	クライアントからかけた負荷の継続時間	35秒間
負荷ツール単体での限界リクエスト数	負荷ツール単体でかけられるリクエスト数の最大値	17,541トランザクション

テスト実施結果

■CPU・ディスク負荷はiostat,vmstatで5秒毎の値を取得

項目	定義	値
Read Latency	読み取り要求を送信し、その応答を受信するまでの合計時間	最小: 2.37秒 最大: 4.59秒 平均: 3.41秒
Read Throughput	1秒あたりの読み取り数	41.8 TPS
Transaction Latency	ネットワーク全体がトランザクションを検証するのにかかる時間	最小 2.52秒 最大 9.36秒 平均 5.88秒
Transaction Throughput	一定期間にブロックチェーンによって有効なトランザクションがコミットされる割合	35.6 TPS
CPU負荷	CPUに対する負荷	CPU使用率 最小: 1% 最大: 5% 平均: 3%
ディスク容量負荷	ディスク容量に対する負荷	1秒当たりのHDDへのwriteの平均 最小: 0.80 KB/s 最大: 146.80 KB/s 平均: 49.47 KB/s

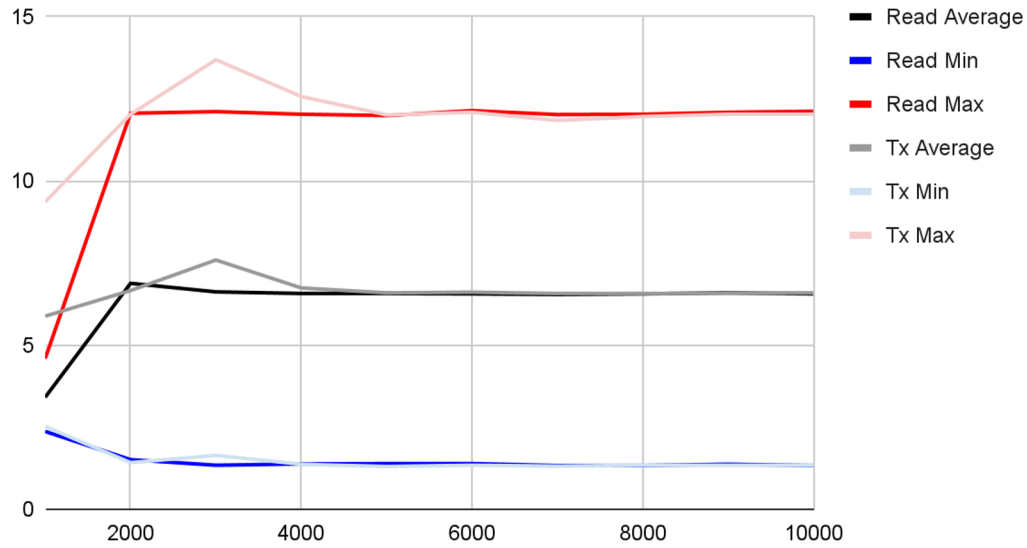
【追加検証1】目標TPSを50 TPSで固定：総トランザクション数を増加させる グラフは次頁に提示

テスト結果

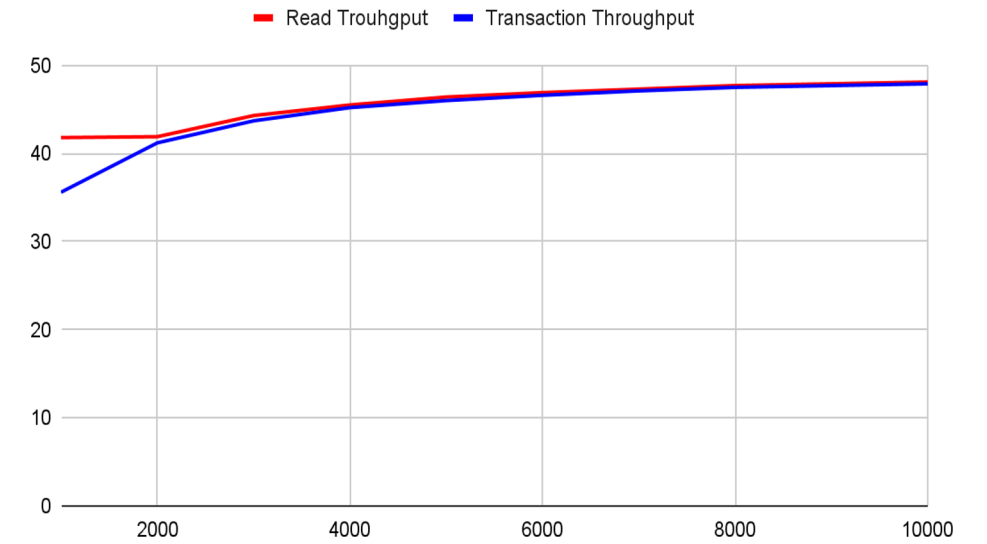
負荷総 リクエスト数	Read Latency	Read Throughput	Transaction Latency	Transaction Throughput	CPU使用率	ディスク容量負荷
1000	最小: 2.37秒 最大: 4.59秒 平均: 3.41秒	41.8 TPS	最小 2.52秒 最大 9.36秒 平均 5.88秒	35.6 TPS	最小: 1% 最大: 5% 平均: 3%	最小: 0.80 KB/s 最大: 146.80 KB/s 平均: 49.47 KB/s
2000	最小: 1.51秒 最大: 12.06秒 平均: 6.88秒	41.9 TPS	最小 1.42秒 最大 12.03秒 平均 6.66秒	41.2 TPS	最小: 2% 最大: 11% 平均: 5.29%	最小: 0.80 KB/s 最大: 408.80 KB/s 平均: 73.55 KB/s
3000	最小: 1.34秒 最大: 12.11秒 平均: 6.62秒	44.3 TPS	最小 1.64秒 最大 13.69秒 平均 7.59秒	43.7 TPS	最小: 1% 最大: 12% 平均: 4.78%	最小: 0.30 KB/s 最大: 496.00 KB/s 平均: 69.37 KB/s
...
9000	最小: 1.37秒 最大: 12.09秒 平均: 6.59秒	47.9 TPS	最小 1.34秒 最大 12.03秒 平均 6.58秒	47.7 TPS	最小: 1% 最大: 15% 平均: 5.96%	最小: 0.80 KB/s 最大: 571.60 KB/s 平均: 87.43 KB/s
10000	最小: 1.33秒 最大: 12.12秒 平均: 6.56秒	48.1 TPS	最小 1.34秒 最大 12.03秒 平均 6.59秒	47.9 TPS	最小: 1% 最大: 16% 平均: 6.41%	最小: 0.80 KB/s 最大: 605.60 KB/s 平均: 109.76 KB/s

※負荷総リクエスト数4000-8000の範囲は紙面の都合割愛

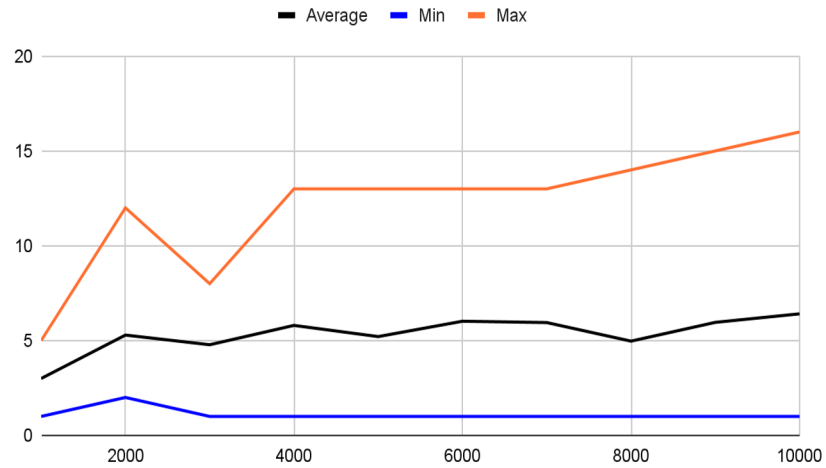
Read Latency / Transaction Latency (秒)



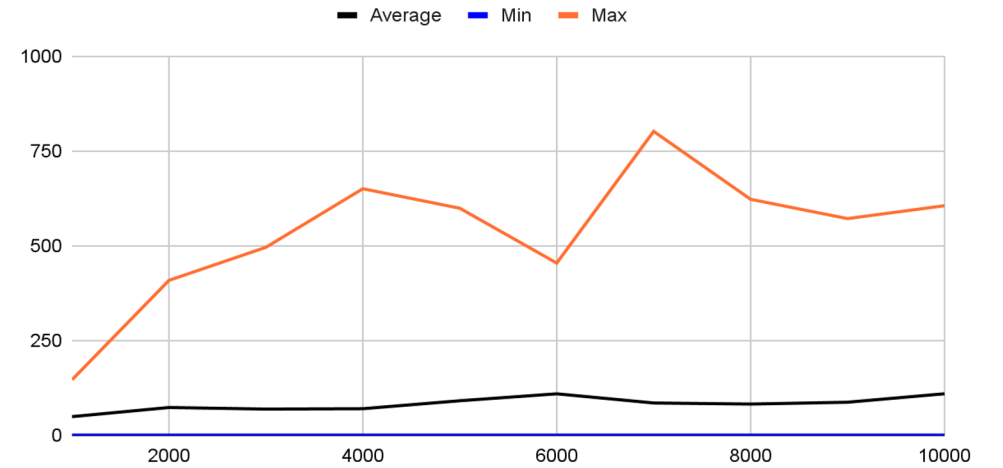
Read Throughput / Transaction Throughput (TPS)



CPU使用率 (%)



ディスク負荷 (KB/s)

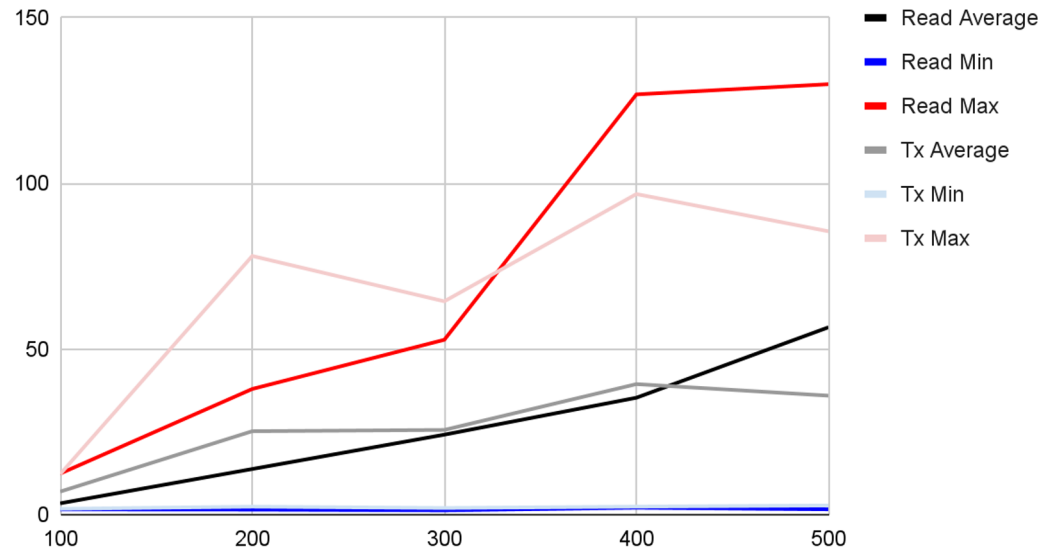


【追加検証2】総トランザクション数を10,000で固定：目標TPSを増加させる グラフは次頁に提示

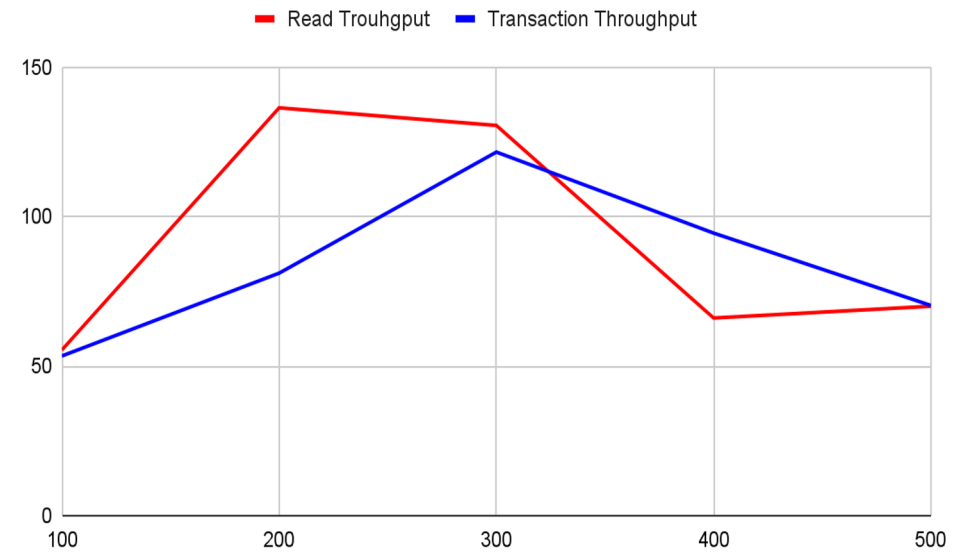
テスト結果

目標TPS	Read Latency	Read Throughput	Transaction Latency	Transaction Throughput	CPU使用率	ディスク容量負荷
100	最小: 1.57秒 最大: 12.39秒 平均: 3.41秒	55.4 TPS	最小 1.75秒 最大 12.23秒 平均 6.95秒	53.4 TPS	最小: 1% 最大: 39% 平均: 17%	最小: 0.8 KB/s 最大: 1093 KB/s 平均: 147.58 KB/s
200	最小: 1.51秒 最大: 37.93秒 平均: 13.77秒	136.4 TPS	最小 2.42秒 最大 78.08秒 平均 25.18秒	81.1 TPS	最小: 1% 最大: 83% 平均: 53.65%	最小: 0.8 KB/s 最大: 1072 KB/s 平均: 125.6 KB/s
300	最小: 1.29秒 最大: 52.78秒 平均: 24.14秒	130.5 TPS	最小 2.01秒 最大 64.39秒 平均 25.57秒	121.6 TPS	最小: 1% 最大: 82% 平均: 56.78%	最小: 0.7 KB/s 最大: 2708 KB/s 平均: 192.75 KB/s
400	最小: 2.09秒 最大: 126.85秒 平均: 35.29秒	66.1 TPS	最小 2.43秒 最大 96.77秒 平均 39.40秒	94.5 TPS	最小: 1% 最大: 82% 平均: 75.42%	最小: 0.7 KB/s 最大: 1894 KB/s 平均: 108.02 KB/s
500	最小: 1.61秒 最大: 129.93秒 平均: 56.55秒	70.3 TPS	最小 2.76秒 最大 85.53秒 平均 35.91秒	70.3 TPS	最小: 1% 最大: 82% 平均: 63.717%	最小: 0.60 KB/s 最大: 2947.2 KB/s 平均: 129.093 KB/s

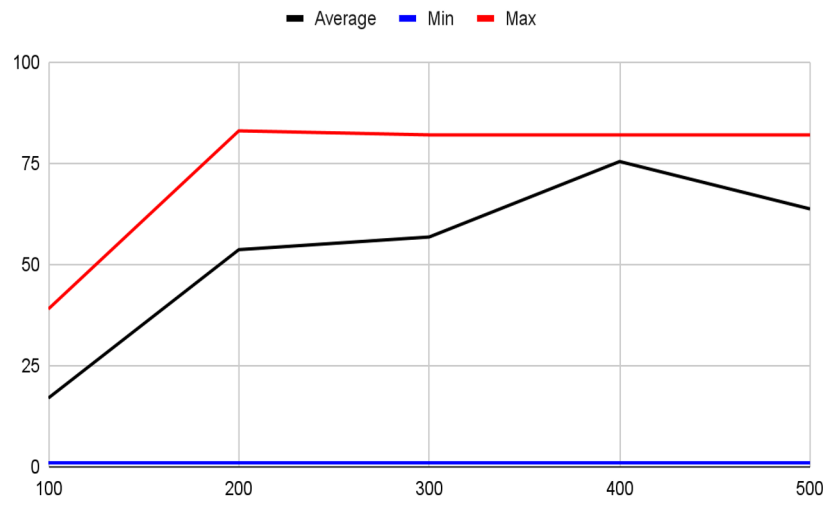
Read Latency / Transaction Latency (秒)



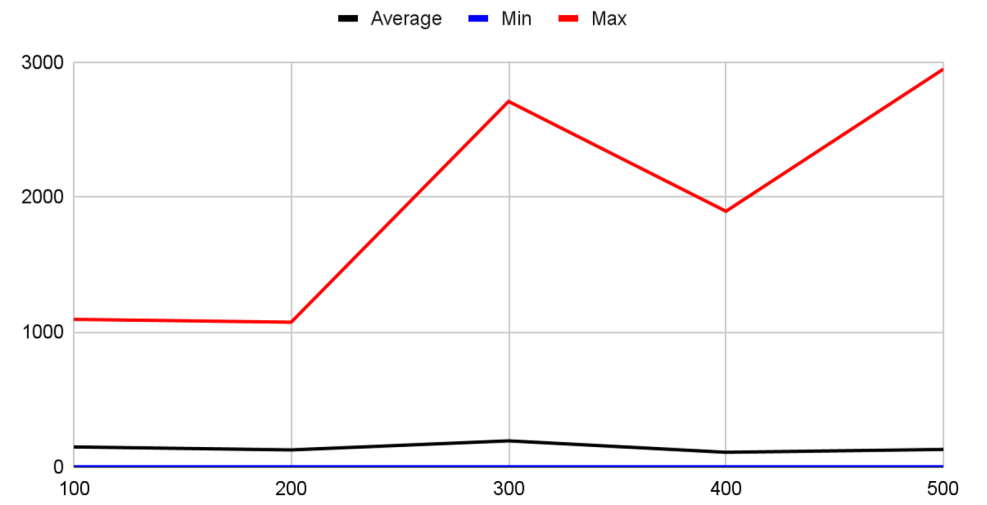
Read Throughput / Transaction Throughput (TPS)



CPU使用率 (%)



ディスク負荷 (KB/s)



負荷総リクエスト数・目標TPSを変数として測定を行ったことから、Gethの性能限界点や可能性までを検証することができた

テスト結果サマリ

1 目標TPSを50 TPSで固定：総トランザクション数を増加させる

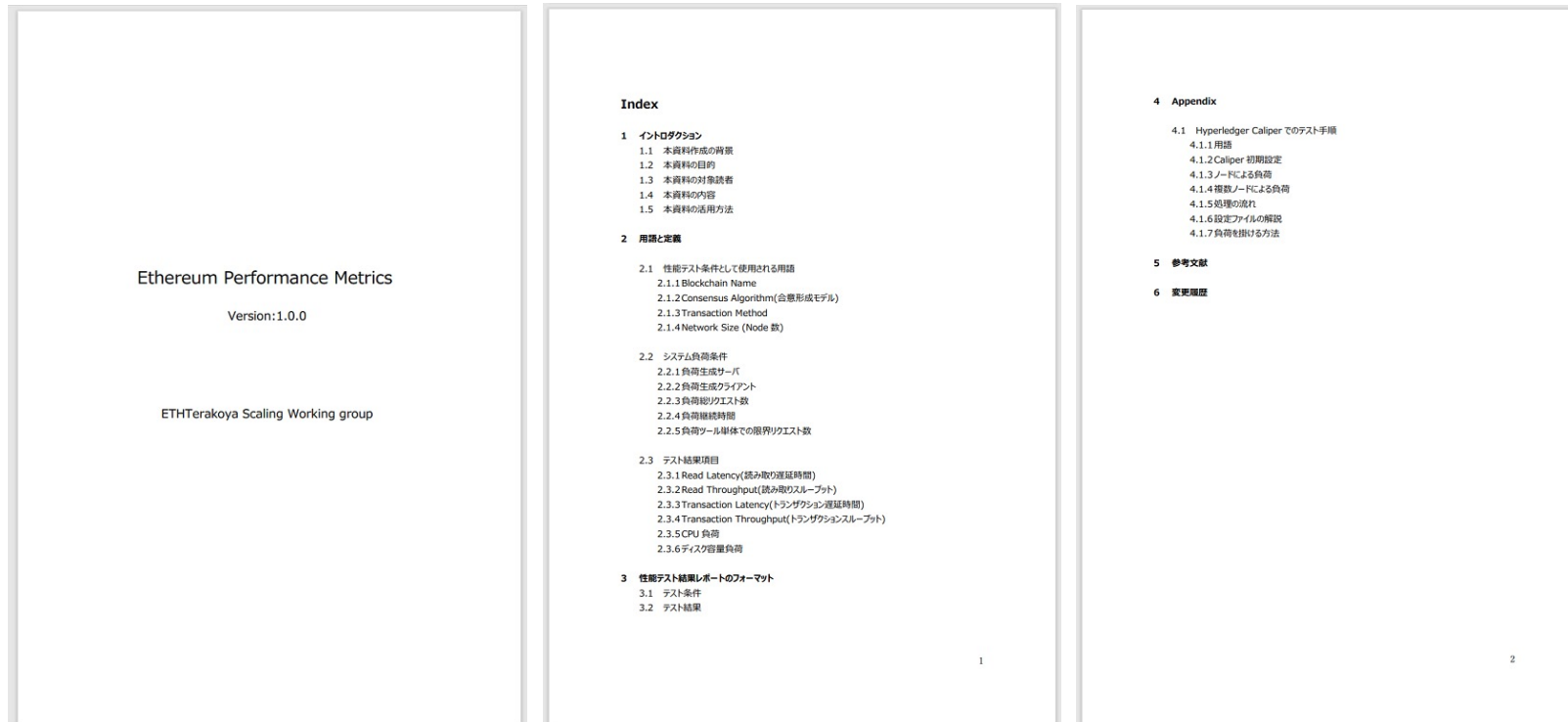
- スループットは総トランザクション数を増やすにつれて50TPSに近づく
- レイテンシーやCPU・ディスク負荷のMAX値は大きくなり、同時にばらつきも大きくなるものの平均値は微増
目標TPSが50TPS固定でリクエストを10,000まで増やしても、通常通りリクエストをさばき負荷はほぼかかっている印象
- ただし、更にトランザクション数を増やすとペンディングトランザクションが蓄積されはじめ、17,000前後で処理が追いつかなくなるため、負荷ツール単体での限界リクエスト数は17,500程度と考えられる

2 総トランザクション数を10,000で固定：目標TPSを増加させる

- スループットは200TPSから300TPSあたりで最大になり、300TPS以降は逆に減少していく
以降CPU使用率とディスク負荷は上昇するが、レイテンシーと負荷が大きくなるだけでスループットが改善しないことから、このあたりが限界値と思われる
- Ethereum mainnetは現在15 TPSほどと言われているので、分散性を犠牲にしてプライベートチェーンにすることによって20倍ほど速度の向上を期待できるといえる

共通の評価指標を持つことで、多岐に渡るチェーンを同条件で検証し、性能比較を行うことが可能となる
ホワイトペーパーへのコメント、およびホワイトペーパーを参照し検証をされた場合はぜひ共有を頂きたい

ホワイトペーパーの公開・更新について



- githubリポジトリ : [xxxxxxx](#)
- ETHTerakoyaWEB : <https://ethereum-terakoya.org/scaling/>
- 問い合わせ先 : ETHTerakoya事務局 contact@ethereum-terakoya.org